**Lesson 9. Strings**

String у Java не є примітивним типом, а це окрім усього іншого означає, що для них також доступний набір функцій, що дозволяє оперувати рядками.

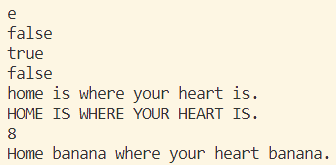
Чому важливо знати як це робити, можемо зрозуміти вже з того, що дуже велика частина даних, які зберігаються у різних застосунках – це сам текстові дані (ім’я користувача, електронна пошта, повідомлення, статус у соцмережі тощо).

**String methods**

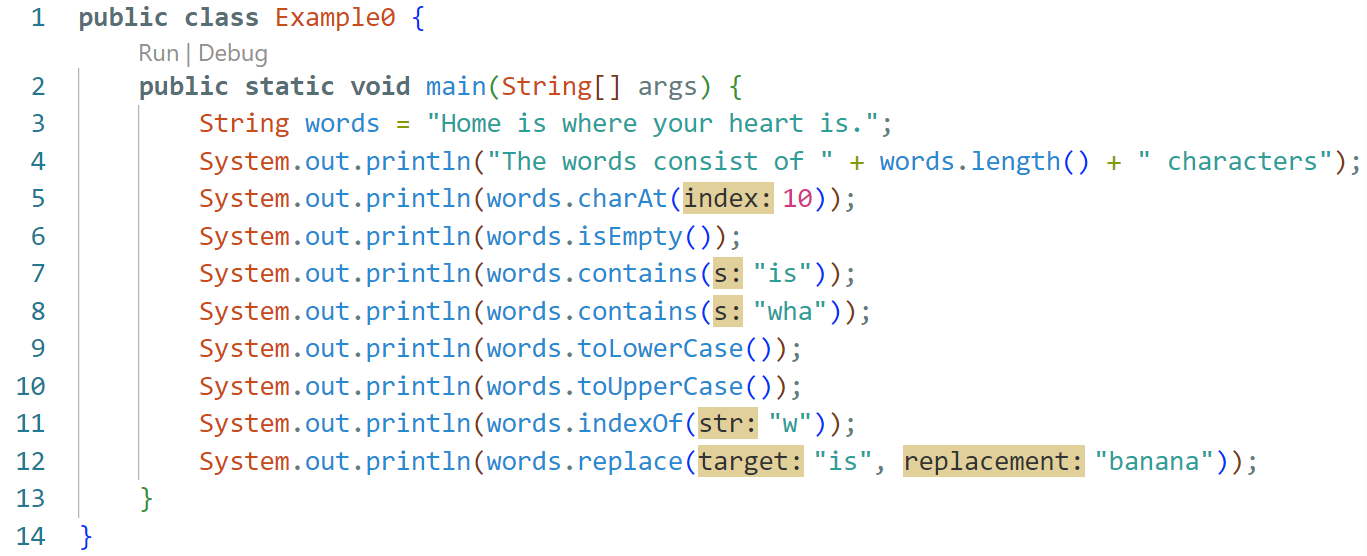
Змінні із типом String зберігають текстові дані, які є нічим іншим, як послідовністю символів (літер та знаків). Кожен символ у послідовності має свій індекс, за яким його можна знайти у рядку. Нумерація індексів починається із 0 (тобто перша літера у рядку має нульовий індекс).

Розглянемо кілька методів (функцій), що дозволяють отримувати інформацію про рядок та виконувати з ним певні дії:

|  |  |  |
| --- | --- | --- |
| Метод | Опис | Тип значення, що повертається |
| length() | Повертає кількість символів у рядку | int |
| isEmpty() | Повертає значення **true** якщо рядок є порожнім, а інакше **false** | boolean |
| charAt(index) | Повертає символ рядка, що знаходиться за вказаним індексом | char |
| contains(s) | Повертає значення **true**, якщо рядок містить вказаний фрагмент | boolean |
| replace(target, replacement) | Повертає нове значення **String**, у якому усі значення target заміщенні на replacement | String |
| toUpperCase() | Повертає нове значення **String**, у якому всі літери нижнього регістру замінено на літери верхнього | String |
| toLowerCase() | Повертає нове значення **String**, у якому всі літери верхнього регістру замінено на літери нижнього | String |
| indexOf(str) | Повертає значення індексу, за яким у рядку буде знайдено фрагмент str (-1 якщо фрагмент не знайдено) | int |
| equals() | Повертає true якщо рядки однакові, а інакше false. | boolean |

**Приклад 1**. Протестуємо нові методи для рядків на простенькому прикладі, аби познайомитись із їх роботою. Результат виконання коду внизу наведено на фрагменті ліворуч.

**Examples**

****

**Tasks**

**Завдання 1**. Створіть програму, що буде отримувати від користувача рядок і робити висновок про те чи рядок порожній. Якщо рядок не порожній, то програма повинна вивести його довжину.

Тестові дані:

*Enter numbers upper bound: 7*

Очікуваний результат:

*0 2 4 6*

Тестові дані:

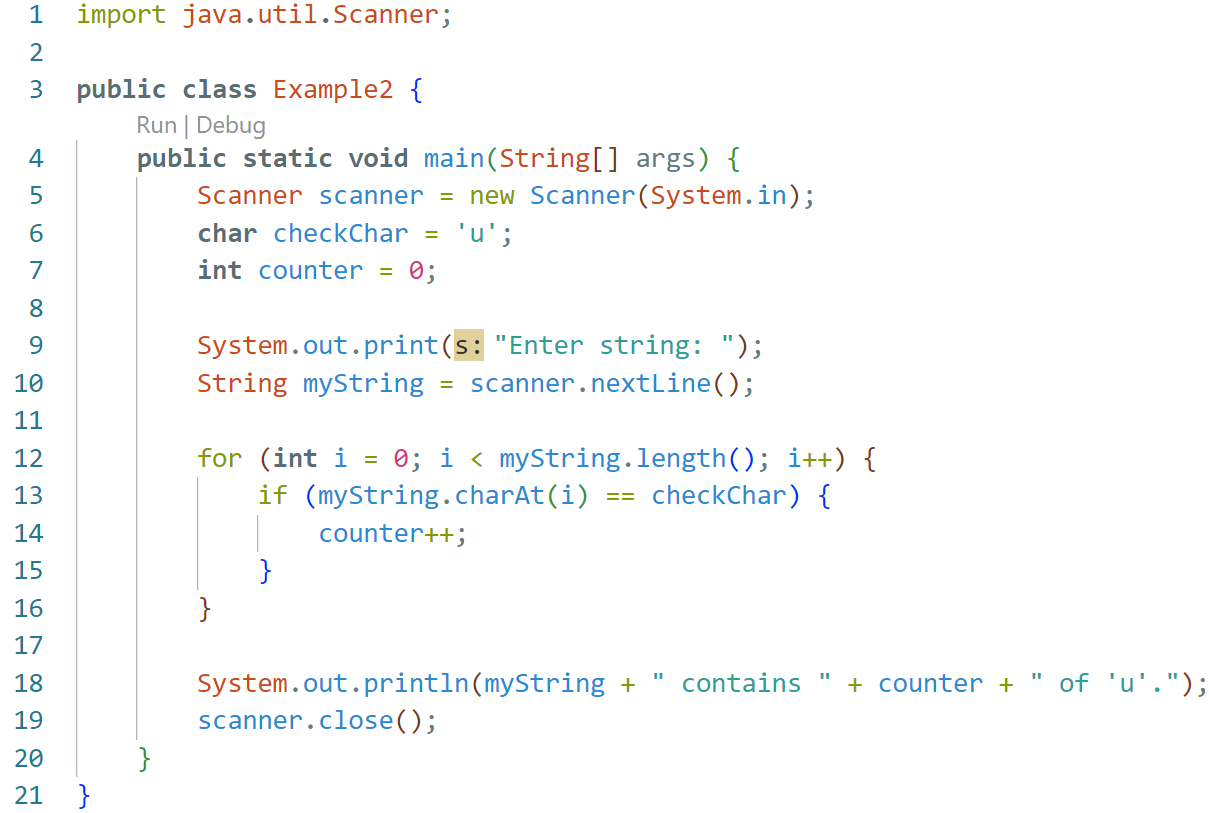
*Enter string:*

*Enter string: cat*

Очікуваний результат:

*The string is empty.*

*The length of the string is: 3*

**Завдання 2**. Створіть програму, що виводить у консоль скільки літер *u* містить введене слово:

**Завдання 3**. Змініть програму таким чином, щоб вона рахувала і великі і маленькі літери ‘u’.

Тестові дані:

*Enter string: Future*

Очікуваний результат:

*Future contains 2 of 'u'.*

**Завдання 4**. Отримавши рядок програма повинна видалити із нього усі голосні і повернути результат.

Тестові дані:

*Enter string: FUture*

Очікуваний результат:

*Future contains 2 of 'u'.*



Тестові дані:

*Entera a word: Deforestation*

Очікуваний результат:

*New string is: Dfrsttn*

**Завдання 5**. Паліндром – слово, яке читається однаково в обидві сторони (наприклад «корок» чи «Анна»). Напишіть програму, яка перевірятиме, чи введений рядок паліндром.

**Homework**

Тестові дані:

*Entera a word: Hobbit*

*Entera a word: Hannah*

Очікуваний результат:

*Word 'Hobbit' is not palindrome.*

*Word 'Hannah' is a palindrome.*

**Завдання 1**. Напишіть програму, що буде приймати від користувача слова, допоки не буде введено слово “tenet”. Коли слово буде введено програма звершується, але перед завершенням виводить слово сформоване із усіх перших літер слів, що було введено під час її виконання.

Тестові дані:

*Enter a word: john*

*Enter a word: apple*

*Enter a word: village*

*Enter a word: antony*

*Enter a word: tenet*

Очікуваний результат:

*Created word is: java*